# Guide de référence rapide pour Debian

Osamu Aoki <osamu\#at\#debian.org>
Traduction en Français : Guillaume Erbs <gerbs\#at\#free.fr>
'Auteurs' page 25

CVS, lun 16 jun 2008 21 :21 :04 UTC

## Résumé

Ce Guide de référence rapide pour Debian (http://qref.sourceforge.net/) se propose de donner une introduction courte au système Debian comme **référence rapide**. C'est un extrait de Guide de référence pour Debian (http://qref.sourceforge.net/).

## Copyright

# Table des matières

# Chapitre 1

# Préface

Mon principe est de **le garder court et simple**.

## 1.1 Conventions du document

Le « Guide de référence rapide pour Debian » procure de l'information par le biais de commandes simples en shell BASH.

Références :
- une page de **manuel Unix** est donnée dans la forme `bash(1)`.
- une page **GNU TEXINFO** est donnée dans la forme `info libc`.

## 1.2 Fondements de la distribution Debian

Debian se présente sous la forme simultanée de 3 éditions :

- stable : Convient pour un serveur de production. Ennuyeux pour une station de travail (ST).
- testing : Convient pour une ST.
- unstable : Ne pas charger cette édition aveuglément.

Lisez au moins la liste de discussion centrale `debian-devel-announce@lists.debian.org` pour des mises à jour sur l'état de Debian. (NdT : Pour les francophones, vous trouverez de l'aide en vous abonnant à la liste de discussion francophone `debian-user-french-request@lists.debian.org`.)

Au mois de mars 2002, ces trois versions correspondent à `Potato` (qualité de production), `Woody` (en beta-test, très stable maintenant), et `Sid` (en alpha-test). En août 2002, juste après la sortie de `Woody`, cela correspond à `Woody` (qualité de production), `Sarge` (en beta-test, elle sera assez difficile pendant quelque temps), et `Sid` (toujours en alpha-test). Lorsque les paquets dans `unstable` n'ont plus de bogues critiques (Release Critical, RC) répertoriés pendant environ une semaine de test, ils sont automatiquement inclus dans `testing`.

En théorie, il y a deux choses que vous pouvez faire pour disposer des dernières versions des applications.

- 'Mise à jour d'une distribution vers `testing`' page suivante (principalement pour des stations de travail)
- 'Porter un paquet vers le système `stable`' page 20 (principalement pour des serveurs)

# Chapitre 2

# Mise à jour d'une distribution vers `testing`

Ce chapitre est basé sur une version plus ancienne de texte original anglais. Veuillez vérifier la version anglaise aussi.

## 2.1 Transition de APT vers la version de Woody

Une mise à jour par le réseau du système APT et de certains paquets principaux vers la version de Woody peut être effectuée comme suit après avoir inclus une source `stable` dans `/etc/apt/sources.list` si vous utilisez toujours Potato.

```
# apt-get update
# apt-get install libc6 perl libdb2 debconf
# apt-get install apt apt-utils dselect dpkg
```

## 2.2 Préparation de la transition (« stable » vers « testing »)

Suivre la version `testing` de Debian a l'effet de bord de recevoir très lentement les correctifs de sécurité. Vous êtes prévenus.

Une mise à jour via le réseau vers « testing » peut être effectuée comme suit (lancer le script suivant go-woody (`http://www.debian.org/doc/manuals/debian-reference/examples/`) pour le faire en une seule commande) :

Videz le fichier `/etc/apt/sources.list` existant

```
# cd /etc/apt
# cp -f sources.list sources.old
# :>sources.list
```

Récupérez une liste propre de dépôts, pour « stable »

```
# cd /
# apt-setup noprobe
 ... sélectionner http ou ftp
```

Ajoutez la section « testing » à cette nouvelle liste. Les lignes deb-src sont commentées.

```
# cd /etc/apt
# grep -e "^deb " sources.list             >sources.deb
# grep -e "^deb-" sources.list             >sources.src
# sed -e "s/stable/testing/"  sources.deb \
     >>sources.list
# sed -e "s/stable/testing/" sources.src | \
     sed -e "s/^deb-/#deb-/"  >>sources.list

# apt-get update
# apt-get install apt apt-utils
# cat >preferences <<EOF
 Package: *
 Pin: release a=testing
 Pin-Priority: 600

 Package: *
 Pin: release a=unstable
 Pin-Priority: 50

 EOF
```

Eventuellement, ajoutez la section « unstable » des archives.

```
# sed -e "s/stable/unstable/" sources.deb \
  >>sources.list
# sed -e "s/stable/unstable/" sources.src | \
 sed -e "s/^deb-/#deb-/"  >>sources.list
```

Voir 'Bases de la gestion de paquets Debian' page 8 pour configurer /etc/apt/sources.list et /etc/apt/preferences.

Vous pouvez maintenant mettre à jour, en utilisant une des méthodes décrites dans la section suivante.

## 2.3 Mise à jour du système Debian

Après avoir configuré correctement /etc/apt/sources.list et /etc/apt/preferences, le système peut être mis à jour vers testing. Voir 'Gestion des paquets Debian' page 7 pour les bases, et 'Résolution de problèmes de mise à jour avec APT' page 13 si vous rencontrez des problèmes.

### 2.3.1 Meilleure méthode de mise à jour avec dselect

Si le système comprend beaucoup de paquets -dev, etc., la méthode suivante avec dselect est recommandée pour un contrôle fin des paquets.

```
# dselect update # à faire avant chaque mise à jour
# dselect select # sélectionne des paquets additionnels
```

Tous vos paquets actuels seront sélectionnés lorsque dselect démarre. dselect peut vous proposer d'autres paquets selon les priorités depends, suggests et recommends. Si vous ne voulez pas ajouter de paquet, tapez simplement Q pour sortir de dselect.

```
# dselect install
```

Vous devrez répondre à quelques questions de configuration des paquets pendant cette partie du processus, aussi ayez vos notes à portée de main et comptez un peu de temps pour cette partie. Voir 'dselect' page 10.

Utilisez dselect. **Il fonctionne toujours :)** Si vous devez mettre à jour sans dselect après Woody, considérez l'utilisation d'aptitude ou d'autres options.

# Chapitre 3

# Gestion des paquets Debian

aptitude est désormais le frontal en mode texte pour APT à utiliser de préférence. Il se rappelle des paquets que vous avez installés volontairement et quels paquets ont été ajoutés par dépendance ; ces derniers sont automatiquement désinstallés par aptitude lorsqu'ils ne sont plus nécessaires à un paquet installé volontairement. Il possède des fonctions avancées de filtrage des paquets mais elles peuvent être difficiles à configurer.

synaptic est désormais le frontal graphique GTK pour APT à utiliser de préférence. Ses fonctions de filtrage des paquets sont plus faciles à utiliser que celles d'aptitude. Il possède aussi un support expérimental pour les Debian Package Tags (http://debtags.alioth.debian.org/) (balises de paquets Debian).

Pour réduire la charge réseau des dépôts Debian et accélérer vos téléchargements, vous devriez récupérer vos paquets depuis les miroirs Debian.

Si vous devez installer le même paquet sur plusieurs machines de votre réseau local, vous pouvez configurer un proxy HTTP local en utilisant squid pour les paquets téléchargés par APT. Si nécessaire, positionnez la variable d'environnement http_proxy ou la variable http dans /etc/apt/apt.conf.

Bien que la fonction de *pinning* d'APT décrite dans apt_preferences(5) soit puissante, ses effets peuvent être difficiles à comprendre et à gérer. Vous devriez la considérer comme une Fonction Avancée.

L'utilisation de chroot est utile pour simultanément assurer à la fois la stabilité du système et l'accès aux dernières versions des programmes.

Ce chapitre est basé sur un système d'après Woody. Certaines fonctions peuvent nécessiter un système Sarge, ou plus récent.

## 3.1 Introduction

Si lire toute la documentation pour les développeurs est trop pour vous, lisez d'abord ce chapitre et commencez à goûter à la puissance de Debian avec testing/unstable :-)

### 3.1.1 Outils de gestion des paquets principaux

```
dpkg      - installation de paquets Debian
apt-get   - frontal pour APT en ligne de commande
aptitude  - frontal avancé pour APT en mode texte et ligne de commande
synaptic  - frontal pour APT en mode graphique GTK
dselect   - gestion des paquets à l'aide de menus
tasksel   - installation de tâches
```

Ces outils ne sont pas tous des alternatives. Par exemple `dselect` utilise à la fois APT et `dpkg`.

APT utilise `/var/lib/apt/lists/*` pour suivre les paquets disponibles tandis que `dpkg` utilise `/var/lib/dpkg/available`. Si vous avez installé des paquets directement en utilisant `aptitude` ou un autre frontal pour APT et que vous voulez utiliser `dselect` pour installer des paquets, assurez-vous de mettre à jour le fichier `/var/lib/dpkg/available` en sélectionnant `[M]ise à jour` dans le menu de `dselect` (ou en exécutant `dselect update`).

`apt-get` récupère automatiquement les paquets dont un paquet demandé dépend. Il n'installe pas les paquets recommandés ou suggérés par le paquet demandé.

`aptitude` au contraire peut être configuré pour installé les paquets recommandés ou suggérés.

`dselect` présente à l'utilisateur une liste de paquets qu'un paquet sélectionné recommande ou suggère et permet de les sélectionner ou pas.

### 3.1.2 Outils pratiques

```
dpkg-reconfigure  - reconfigure un paquet déjà installé
                    (s'il utilise debconf)
dpkg-source       - gère les paquets sources
dpkg-buildpackage - automatise la création d'un paquet
apt-cache         - recherche un paquet dans le cache local
```

## 3.2 Bases de la gestion de paquets Debian

### 3.2.1 Configurer APT

Configurer `sources.list` comme décrit dans 'Préparation de la transition (« stable » vers « testing »)' page 3. [1]

[1] Si vous suivez `testing` ou `unstable` vous pouvez enlever les références à `stable` de `sources.list` et `/etc/apt/preferences` parce que `testing` démarre à partir d'une copie de `stable`.

### 3.2.2 Installation de tâches

Vous pouvez installer des ensembles de paquets habituellement nécessaires pour certaines utilisations d'un système Debian. Ces ensembles de paquets sont appelés « tâches ».

Le moyen le plus simple d'installer des tâches lors de l'installation initiale est d'utiliser `tasksel`. Notez que vous devez exécuter

```
dselect update
```

avant de l'utiliser.

`aptitude` permet aussi d'installer des tâches et c'est l'outil recommandé pour cela. Il permet de désélectionner des paquets dans les tâches avant de procéder à l'installation.

### 3.2.3 `aptitude`

`aptitude` est un programme d'installation de paquets par sélection dans des menus, similaire à `dselect` mais développé de zéro au-dessus d'APT. Il peut être utilisé comme alternative à `apt-get` pour la plupart des commandes. Voir `aptitude(1)` et `file:///usr/share/doc/aptitude/README`.

`aptitude` en plein écran accepte des commandes à une touche, généralement en minuscule.

```
Touche   Action
F10      Menu
?        Aide (listing complet)
u        Mise à jour des informations de l'archive de paquets
+        Marque un paquet pour mise-à-jour ou installation
-        Marque un paquet pour suppression (garde la configuration)
_        Marque un paquet pour purge (supprime la configuration)
=        Place un paquet en maintien
U        Marque tous les paquets qui peuvent être mis à jour
g        Téléchargement et installation des paquets sélectionnés
q        Sortie de l'écran courant et sauvegarde des changements
x        Sortie de l'écran courant sans sauvegarde
Enter    Visualisation d'information sur un paquet
C        Visualisation des changements d'un paquet
|        Change la limite des paquets affichés
/        Recherche la première occurence
\        Répète la dernière recherche
```

Comme `apt-get`, `aptitude` installe les dépendances d'un paquet demandé. `aptitude` offre aussi une option pour récupérer les paquets qui sont recommandés ou suggérés par un paquet à installer. Vous pouvez changer ce comportement en choisissant `F10 -> Options -> Dependency handling` dans le menu.

Autres avantages d'`aptitude` :

- `aptitude` offre accès à toutes les versions d'un paquet.
- `aptitude` enregistre toutes ses actions dans `/var/log/aptitude`.
- `aptitude` rend facile le suivi des logiciels obsolètes en les listant dans « *Obsolete and Locally Created Packages* ».
- `aptitude` inclut un système de recherche puissant pour trouver des paquets particuliers ou limité l'affichage des paquets. Les utilisateurs familiers avec `mutt` seront rapidement à l'aise, puisque mutt a inspiré la syntaxe des expressions. Voir « *SEARCHING, LIMITING, AND EXPRESSIONS* » dans `file:///usr/share/doc/aptitude/README`.
- `aptitude` en plein écran intègre `su` et peut être utilisé par un utilisateur normal jusqu'à ce qu'il y ait réellement besoin des privilèges de l'administrateur.

### 3.2.4 `dselect`

Dans les versions stables jusqu'à Potato inclus, `dselect` était l'outil de gestion des paquets principal. Pour Sarge, il est recommandé d'utiliser `aptitude` à la place.

Quand `dselect` est lancé, il sélectionne automatiquement tous les paquets « Required », « Important » et « Standard ».

`dselect` a une interface utilisateur déroutante. La plupart des utilisateurs s'y font cependant. Il a quatre commandes (en majuscules !) :

```
Touche      Action
Q           Quitter. Confirme la sélection courante et quitte
            (passe outre les dépendances).
R           Rétablir. Je ne voulais pas faire ça.
D           (Damn it!) Je me fiche de ce que dselect pense. Fais comme ça !
U           Positionne tout dans l'état sUggéré.
```

Avec `D` et `Q`, vous pouvez sélectionner des paquets en conflit à vos risques et périls. Faites attention en utilisant ces commandes.

Ajoutez une ligne « expert » dans `/etc/dpkg/dselect.cfg` pour réduire la quantité d'informations affichée.

Si votre machine exécute `dselect` lentement, vous pouvez lancer `dselect` sur une machine plus rapide pour trouver les paquets et utiliser `apt-get install` sur la machine lente pour les installer.

### 3.2.5 Suivre une distribution avec APT

Pour suivre la distribution `testing` au fur et à mesure de ses changements, votre fichier `/etc/apt/preferences` doit ressembler à cela :

```
Package: *
```

```
Pin: release a=testing
Pin-Priority: 800

Package: *
Pin: release a=stable
Pin-Priority: 600
```

Attention : suivre la version testing de Debian a l'effet de bord de recevoir très lentement les correctifs de sécurité. Ces paquets sont envoyés dans unstable et migrent vers testing seulement après un certain délai.

Voir apt_preferences(5) pour des exemples plus compliqués qui vous permettront, par exemple, de suivre testing tout en installant certains paquets de unstable.

Des exemples de /etc/apt/preferences qui bloquent certains paquets importants dans leur version la plus stable en suivant la version moins stable d'autres paquets non essentiels sont disponibles dans le sous-répertoire d'exemples (http://www.debian.org/doc/manuals/debian-reference/examples/) dans les fichiers preferences.testing et preferences.unstable.

Si vous mélangez les versions de Debian, testing dans stable ou unstable dans stable, vous pourriez récupérer sans le vouloir des paquets importants, comme libc6, de testing ou unstable et il n'y a aucune garantie qu'ils ne soient pas bogués. Vous êtes prévenus.

Aussi, le fichier preferences.stable force tous les paquets à faire une mise à niveau inférieur vers stable.

La mise à niveau inférieur depuis une version plus récente d'un **paquet** n'est pas supportée officiellement dans Debian. Cependant, vous pouvez avoir à mettre à niveau inférieur un paquet spécifique pour réinstaller une version d'un paquet qui marche lorsque la nouvelle version ne fonctionne pas. Vous pouvez trouver ces paquets précédents localement dans /var/cache/apt/archives/ ou à distance sur http://snapshot.debian.net/. Voir aussi 'Sauvetage avec dpkg' page 14.

La mise à niveau inférieur depuis une version plus récente d'une **distribution** n'est pas non plus supportée officiellement et risque de poser des problèmes. Cependant, cela peut valoir le coup d'essayer en dernier recours si vous êtes désespérés.

### 3.2.6 Commandes aptitude, apt-get et apt-cache

Pour fonctionner avec testing comme décrit dans l'exemple précédent, le système peut être géré avec les commandes suivantes :

– aptitude upgrade (ou apt-get upgrade ou aptitude dist-upgrade ou apt-get dist-upgrade)
  Cela suit la distribution testing et met à jour tous les paquets du système en installant leurs dépendances de testing. [2]

[2] La différence entre upgrade et dist-upgrade n'apparaît que lorsque de nouvelles versions de paquets ont

– `apt-get dselect-upgrade`
  Cela suit la distribution `testing` et met à jour tous les paquets du système avec la sélection de `dselect`.
– `aptitude install` *`package`*`/unstable`
  Cela installe *package* de `unstable` en prenant les dépendances dans `testing`.
– `aptitude install -t unstable` *`package`*
  Cela installe *package* de `unstable` en prenant les dépendances aussi dans `unstable` en mettant `Pin-Priority` de `unstable` à 990.
– `apt-cache policy` *`foo bar ...`*
  Cela affiche l'état des paquets *foo bar ...*.
– `aptitude show` *`foo bar ...`* `| less` (ou `apt-cache show` *`foo bar ...`* `| less`)
  Cela affiche l'information sur les paquets *foo bar ...*.
– `aptitude install` *`foo=2.2.4-1`*
  Cela installe la version *2.2.4-1* du paquet *foo*.
– `aptitude install` *`foo bar-`*
  Cela installe la paquet *foo* et supprime le paquet *bar*.
– `aptitude remove` *`bar`*
  Cela supprime le paquet *bar* mais garde ses fichiers de configuration.
– `aptitude purge` *`bar`*
  Cela supprime le paquet *bar* et tous ses fichiers de configuration.

Dans les exemples précédents, l'utilisation de l'option `-u` permet qu'`apt-get` affiche la liste des paquets qui seront mis à jour et demande à l'utilisateur de confirmer avant de continuer. Ce qui suit permet de configurer l'option `-u` par défaut :

```
$ cat >> /etc/apt/apt.conf << .
// Always show packages to be upgraded (-u)
APT::Get::Show-Upgraded "true";
.
```

Utilisez l'option `--no-act` pour simuler des actions sans les effectuer.

## 3.3 Commandes de survie Debian

Avec ces connaissances, on peut vivre une vie d'éternelle mise à jour :-)

### 3.3.1 Rechercher les bogues de Debian et demander de l'aide

Si vous avez des problèmes avec un paquet, consultez ces sites en premier avant de demander de l'aide ou de remplir un rapport de bogue. (`lynx`, `links` et `w3m` fonctionnent bien) :

---

des dépendances différentes des anciennes versions. Voir `apt-get(8)` pour plus de détails. `aptitude upgrade` et `aptitude dist-upgrade` lancent `aptitude` en ligne de commande. Vous pouvez changer pour le mode plein écran en appuyant sur `e`.

```
$ lynx http://bugs.debian.org/
$ lynx http://bugs.debian.org/package-name  # si vous connaissez
                                          # le nom du paquet
$ lynx http://bugs.debian.org/bugnumber     # si vous connaissez
                                          #le numéro du bogue
```

Utilisez le moteur de recherche Google (www.google.com) en incluant « site :debian.org » comme mot-clé.

En cas de doute, lisez la documentation. Positionnez `CDPATH` comme suit :

```
export CDPATH=.:/usr/local:/usr/share/doc
```

et tapez

```
$ cd packagename
$ pager README.Debian # s'il existe
$ mc
```

### 3.3.2 Résolution de problèmes de mise à jour avec APT

Des problèmes de dépendances peuvent arriver lors de la mise à jour avec `unstable/testing`, comme décrit dans 'Mise à jour du système Debian' page 5. La plupart du temps, c'est parce qu'un paquet qui doit être mis à jour possède une dépendance qui n'est pas trouvée. Ces problèmes sont résolus en utilisant

```
# aptitude dist-upgrade
```

Si cela ne marche pas, répétez les opérations suivantes jusqu'à ce que le problème se résolve tout seul :

```
# aptitude -f upgrade         # continue « upgrade » malgré les erreurs
... ou
# aptitude -f dist-upgrade    # continue « dist-upgrade » malgré les erreurs
```

De mauvais scripts de mise à jour peuvent causer des problèmes récurrents. Il est recommandé de résoudre ce genre de situations en consultant les scripts `/var/lib/dpkg/info/packagename.{post,pre}{inst,rm}` du paquet en question et de lancer :

```
# dpkg --configure -a    # configure les paquets partiellement installés
```

Si un script se plaint d'un fichier de configuration manquant, cherchez dans /etc le fichier de configuration correspondant. Si un fichier de configuration avec une extension .dpkg-new (ou quelquechose similaire) existe, renommez-le (mv) en un fichier sans suffixe.

Des problèmes de dépendances peuvent survenir en installant des paquets de unstable/testing. Il existe des moyens de passer outre les dépendances.

```
# aptitude -f install package # passe outre les dépendances non résolues
```

Une autre méthode pour résoudre ces problèmes est d'utiliser le paquet equivs. Voir 'Le paquet equivs' page 23. Voir file:///usr/share/doc/equivs/README.Debian.

### 3.3.3 Sauvetage avec dpkg

Si vous n'arrivez plus à utiliser APT vous pouvez télécharger les paquets sur un miroir Debian et les installer en utilisant dpkg. Si vous n'avez pas accès au réseau, vous pouvez chercher des copies des paquets dans /var/cache/apt/archives/.

```
# dpkg -i fetchmail_6.2.5-4_i386.deb
```

Si l'installation d'un paquet de cette façon échoue à cause d'une dépendance non résolue, et que vous avez vraiment besoin d'installer le paquet, vous pouvez ignorer les dépendances en utilisant --ignore-depends, --force-depends et d'autres options avec dpkg. Voir See dpkg(8) pour plus de détails.

### 3.3.4 Récupérer les données de sélection des paquets

Si /var/lib/dpkg/status est corrompu, le système Debian perd les données de sélection des paquets et en est grandement affecté. Cherchez le vieux fichier /var/lib/dpkg/status dans /var/lib/dpkg/status-old ou /var/backups/dpkg.status.*.

Garder /var/backups/ dans une partition séparée peut être une bonne idée puisque ce répertoire contient beaucoup de données système importantes.

Si le vieux fichier /var/lib/dpkg/status n'est pas disponible, vous pouvez quand même récupérer les données depuis des répertoires de /usr/share/doc/.

```
# ls /usr/share/doc | \
  grep -v [A-Z] | \
  grep -v '^texmf$' | \
  grep -v '^debian$' | \
  awk '{print $1 " install"}' | \
  dpkg --set-selections
# dselect --expert # réinstallez le système, utilisez dselect si besoin
```

### 3.3.5 Sauver un système après avoir perdu `/var`

Comme le répertoire `/var` contient des données mises à jour régulièrement, comme le mail, il est plus susceptible de corruption que, par exemple `/usr/`. Le placer dans une partition indépendante limite les risques. Si le pire arrive, vous pouvez avoir à reconstruire le répertoire `/var` pour sauver le système Debian.

Obtenez la structure du répertoire `/var` à partir d'un système Debian qui fonctionne, basé sur la même ou une plus ancienne version de Debian, par exemple `/var`, et placez la à la racine du système cassé. Ensuite

```
# cd /
# mv var var-old      # s'il reste des données utiles
# tar xvzf var.tar.gz # utilisez la structure de Woody
# aptitude            # ou dselect
```

Cela doit fournir un système qui fonctionne. Vous pouvez récupérer les sélections de paquets en utilisant la technique décrite à 'Récupérer les données de sélection des paquets' page précédente. ([FIXME] : Cette procédure nécessite plus de tests.)

### 3.3.6 Installer un paquet sur un système qui ne démarre plus

Démarrez Linux en utilisant une disquette/CD de sauvetage Debian ou une autre partition sur un système Linux multi-boot. Montez le système qui ne démarre plus sur `/target` et utilisez le mode d'installation chroot de `dpkg`.

```
# dpkg --root /target -i packagefile.deb
```

Configurez ensuite et résolvez les problèmes.

Au fait, si c'est seulement un `lilo` cassé qui empêche le démarrage, vous pouvez démarrer avec le disque de sauvetage standard Debian. A l'invite de démarrage, en considérant que la partition racine de Linux est sur `/dev/hda12` et que vous souhaitez le niveau de fonctionnement 3, entrez :

```
boot: rescue root=/dev/hda12 3
```

Vous avez maintenant démarré un système presque complètement fonctionnel avec le noyau de la disquette. (Il peut y avoir de petits problèmes dus au manque d'options du noyau ou de modules.)

### 3.3.7 Que faire si dpkg est cassé

Cela pose de sérieux problèmes pour l'installation des fichiers .deb. Les commandes suivantes vous guident pour récupérer cette situation. (Sur la première ligne, vous pouvez remplacer « links » par votre navigateur favori.)

```
$ links http://http.us.debian.org/debian/pool/main/d/dpkg/
  ... téléchargez le bon dpkg_version_arch.deb
$ su
password: *****
$ ar x dpkg_version_arch.deb
# mv data.tar.gz /data.tar.gz
# cd /
# tar xzfv data.tar.gz
```

Pour une machine i386, http://packages.debian.org/dpkg peut aussi être utilisé comme URL.

## 3.4 Commandes « nirvana » de Debian

A la **lumière** de ces commandes, on sera sauvé de l'éternelle lutte karmique contre l'enfer de la mise à jour afin d'atteindre le **nirvana** Debian. :-)

### 3.4.1 Informations sur un fichier

De l'information sur un fichier peut être trouvée avec :

```
$ dpkg {-S|--search} pattern # cherche « pattern »
                             # parmi les paquets installés
$ wget http://ftp.us.debian.org/debian/dists/sarge/Contents-i386.gz
$ zgrep -e pattern Contents-i386.gz
             # cherche « pattern » parmi les fichiers de l'archive Debian
```

Sinon, utilisez les commandes spécialisées pour les paquets :

```
# aptitude install dlocate
$ dlocate filename
         # solution de rechange rapide pour dpkg -L et dpkg -S
...
# aptitude install auto-apt # outil d'installation de paquets à la demande
# auto-apt update          # crée le fichier db pour auto-apt
$ auto-apt search pattern
         # recherche exhaustive dans l'archive de paquets, installés ou non
```

### 3.4.2 Informations sur un paquet

Cherche et affiche les informations sur un paquet. Assurez-vous qu'apt pointe vers la bonne archive en éditant /etc/apt/sources.list. Si vous voulez voir les différences entre un paquet de testing/unstable et un paquet installé, utilisez apt-cache policy—plutôt sympa.

```
# apt-get   check          # met à jour le cache et vérifie les dépendances
$ apt-cache search  texte  # cherche un paquet à partir de "texte"
$ apt-cache policy  paquet # information sur la priorité d'un paquet
$ apt-cache show -a paquet # affiche la description d'un paquet
                             dans toutes les distributions
$ apt-cache showpkg paquet # informations de debogage sur un paquet
# dpkg  --audit|-C         # cherche les paquets partiellement installés
$ dpkg {-s|--status} paquet ... # état et description d'un paquet installé
$ dpkg -l paquet ...       # état du paquet installé (1 ligne)
$ dpkg -L paquet ...       # liste les noms des fichiers installés
                             par le paquet
```

apt-cache showsrc n'est pas documenté à la sortie de Woody, mais fonctionne :)

Vous pouvez aussi trouver des informations sur les paquets (j'utilise mc pour naviguer) dans :

```
/var/lib/apt/lists/*
/var/lib/dpkg/available
```

La comparaison des fichiers suivants fournit des informations sur ce qu'il s'est exactement passé lors des dernières installations.

```
/var/lib/dpkg/status
/var/backups/dpkg.status*
```

### 3.4.3 Installation automatique avec APT

Pour une installation non interactive, ajoutez la ligne suivante dans /etc/apt/apt.conf :

```
Dpkg::Options {"--force-confold";}
```

C'est l'équivalent de lancer aptitude -y install *packagename* ou apt-get -q -y install *packagename*. Parce que cela répond automatiquement « oui » à toutes les questions, cela peut causer des problèmes, aussi faites attention en utilisant cette astuce. Voir apt.conf(5) et dpkg(1).

Vous pouvez configurer les paquets par la suite en suivant 'Reconfigurer les paquets installés' page suivante.

### 3.4.4 Reconfigurer les paquets installés

Utilisez les commandes suivantes pour reconfigurer un paquet installé.

```
# dpkg-reconfigure --priority=medium package [...]
# dpkg-reconfigure --all   # reconfigure tous les paquets
# dpkg-reconfigure locales # génère de nouvelles locales
# dpkg-reconfigure --p=low xserver-xfree86 # reconfigure le serveur X
```

Faites cela pour `debconf` si vous avez besoin de changer le mode de dialogue de `debconf` de façon permanente.

Certains logiciels sont livrés avec des scripts de configuration. [3]

```
apt-setup     - crée /etc/apt/sources.list
install-mbr   - installe un gestionnaire de Master Boot Record
tzconfig      - configure le fuseau horaire local
gpmconfig     - configure le gestionnaire de souris gpm
eximconfig    - configure Exim (MTA)
texconfig     - configure teTeX
apacheconfig  - configure Apache (httpd)
cvsconfig     - configure CVS
sndconfig     - configure le système sonore
...
update-alternatives - configure la commande par défaut ;
                      par exemple, vim pour vi
update-rc.d         - gestion des scripts de démarrage System-V
update-menus        - système de menus Debian
...
```

### 3.4.5 Enlever et purger des paquets

Enlever un paquet en gardant la configuration :

```
# aptitude remove package ...
# dpkg  --remove package ...
```

Enlever un paquet et sa configuration :

```
# aptitude purge package ...
# dpkg    --purge        package ...
```

[3] Certains scripts de configuration disparaissent dans les dernières versions de Sarge et les fonctions de configuration des paquets sont déplacées dans le système `debconf`.

### 3.4.6 Garder de vieux paquets

Par exemple, pour garder `libc6` et `libc6-dev` lors de l'utilisation de `dselect` et `aptitude install` *`paquet`*, on peut utiliser les commandes suivantes :

```
# echo -e "libc6 hold\nlibc6-dev hold" | dpkg --set-selections
```

`aptitude install` *`paquet`* ne sera pas gêné par cela. Pour garder un paquet en forçant une mise à niveau inférieur automatique avec `aptitude upgrade` *`paquet`* ou `aptitude dist-upgrade`, ajoutez ce qui suit dans `/etc/apt/preferences` :

```
Package: libc6
Pin: release a=stable
Pin-Priority: 2000
```

Ici, l'entrée « `Package:` » ne peut pas utiliser d'entrées comme « `libc6*` ». Si vous voulez garder tous les paquets binaires liés au paquet source `glibc` synchronisés, il faut les lister explicitement.

La commande suivante liste les paquets à garder :

```
dpkg --get-selections "*"|grep -e "hold$"
```

### 3.4.7 Système mixte `stable/testing/unstable`

`apt-show-versions` peut lister les versions des paquets disponibles dans chaque distribution.

```
$ apt-show-versions | fgrep /testing | wc
... nombre de paquets de testing
$ apt-show-versions -u
... liste de paquets qu'il est possible de mettre à jour
$ aptitude install `apt-show-versions -u -b | fgrep /unstable`
... met à jour tous les paquets de unstable à leur dernière version
```

### 3.4.8 Supprimer les paquets du cache

Les installations de paquets avec APT laissent des fichiers de paquets dans le cache `/var/cache/apt/archives` ; ce cache doit être nettoyé.

```
# aptitude autoclean # efface les fichiers de paquets inutiles
# aptitude clean     # efface tous les fichiers de paquets du cache
```

### 3.4.9 Enregistrer/copier la configuration du système

Pour faire une copie locale de l'état de la sélection des paquets :

```
# debconf-get-selections > debconfsel.txt
# dpkg --get-selections "*" >myselections   # ou utiliser \*
```

"*" inclut aussi les noms des paquets à purger dans `myselections`.

Vous pouvez transférer ce fichier vers un autre ordinateur, et l'installer avec :

```
# dselect update
# debconf-set-selections < debconfsel.txt
# dpkg --set-selections <myselections
# apt-get -u dselect-upgrade    # ou dselect install
```

### 3.4.10 Porter un paquet vers le système `stable`

Pour des mises à jour partielles du système `stable`, recréer un paquet pour l'environnement utilisé à partir des sources est recommandé. Cela évite de nombreuses mises à jour de paquets à cause des dépendances. D'abord, ajouter l'entrée suivante dans `/etc/apt/sources.list` :

```
deb-src http://http.us.debian.org/debian testing \
 main contrib non-free
deb-src http://http.us.debian.org/debian unstable \
 main contrib non-free
```

Chaque entrée de `deb-src` est ici scindée en deux lignes à cause de contraintes d'impression, mais les entrées réelles dans `sources.list` doivent tenir en une seule ligne.

Ensuite, récupérez les sources et construisez un paquet :

```
$ apt-get update  # met à jour la liste des paquets source
$ apt-get source package
$ dpkg-source -x package.dsc
$ cd package-version
  ... cherche parmi les paquets requis (Build-Depends dans le fichier .dsc)
      et les installe. Il faut le paquet « fakeroot » aussi.

$ dpkg-buildpackage -rfakeroot

  ...ou (si pas de signature)
$ dpkg-buildpackage -rfakeroot -us -uc # utilisez « debsign »
                                         ensuite si nécessaire
```

```
  ...ensuite pour installer
$ su -c "dpkg -i packagefile.deb"
```

Normalement, il n'y a besoin d'installer que peu de paquets avec un suffixe « -dev » pour satisfaire les dépendances. `debsign` est dans le paquet `devscripts`. `auto-apt` peut aider à résoudre les dépendances. L'utilisation de `fakeroot` évite une utilisation non nécessaire du compte root.

Dans Woody, ces problèmes de dépendances peuvent être simplifiés. Par exemple, pour compiler le paquet source de `pine` :

```
# apt-get build-dep pine
# apt-get source -b pine
```

### 3.4.11 Archive locale de paquets

Afin de créer une archive locale de paquets qui soit compatible avec les systèmes APT et dselect, un fichier `Packages` doit être créé et les fichiers de paquets ont besoin d'être archivés dans un arbre de répertoires particulier.

Un dépôt local similaire à une archive officielle Debian peut être créé avec :

```
# aptitude install dpkg-dev
# cd /usr/local
# install -d pool # les paquets physiques sont placés ici
# install -d dists/unstable/main/binary-i386
# ls -1 pool | sed 's/_.*$/ priority section/' | uniq > override
# editor override # ajustez priority et section
# dpkg-scanpackages pool override /usr/local/ \
   > dists/unstable/main/binary-i386/Packages
# cat > dists/unstable/main/Release << EOF
Archive: unstable
Version: 3.0
Component: main
Origin: Local
Label: Local
Architecture: i386
EOF
# echo "deb file:/usr/local unstable main" \
   >> /etc/apt/sources.list
```

Autrement, un dépôt local peut être créé de façon rapide et sale :

```
# aptitude install dpkg-dev
# mkdir /usr/local/debian
# mv /some/where/package.deb /usr/local/debian
# dpkg-scanpackages /usr/local/debian /dev/null | \
  gzip - > /usr/local/debian/Packages.gz
#  echo "deb file:/usr/local/debian ./" >> /etc/apt/sources.list
```

Ces archives peuvent être accédées à distance en fournissant ces répertoires aux méthodes HTTP ou FTP des entrées du fichier `/etc/apt/sources.list`.

### 3.4.12 Convertir ou installer un paquet non Debian

`alien` permet de convertir des paquets binaires fournis aux formats `rpm` de Red Hat, `slp` de Stampede, `tgz` de Slackware, et `pkg` de Solaris en paquet `deb` de Debian. Si vous voulez utiliser un paquet d'une autre distribution Linux, vous pouvez utiliser `alien` pour le convertir vers votre format de paquet préféré et l'installer. `alien` supporte aussi les paquets LSB.

### 3.4.13 Installation automatique

`auto-apt` est une commande d'installation de paquets à la demande.

```
$ sudo auto-apt update
 ... met à jour la base de données
$ auto-apt -x -y run
Entering auto-apt mode: /bin/bash
Exit the command to leave auto-apt mode.
$ less /usr/share/doc/med-bio/copyright # accède un fichier non existant
 ... installe le paquet qui fournit ce fichier
 ... installe aussi les dépendances
```

### 3.4.14 Vérifier les fichiers de paquets installés

`debsums` permet la vérification des fichiers de paquets installés grâce à des sommes de contrôle MD5. Certains paquets n'ont pas de somme de contrôle MD5. Une solution temporaire pour les administrateurs système :

```
# cat >>/etc/apt/apt.conf.d/90debsums
DPkg::Post-Install-Pkgs {"xargs /usr/bin/debsums -sg";};
^D
```

par Joerg Wendland <`joergland@debian.org`> (non testé).

## 3.5 Autres particularités de Debian

### 3.5.1 La commande `dpkg-divert`

`dpkg-divert` force `dpkg` à ne pas installer un fichier à sa place par défaut, mais à une place **détournée**. Les **détournements** peuvent être utilisés dans les scripts de paquets Debian pour déplacer un fichier qui crée un conflit. Les administrateurs système peuvent utiliser ceci pour recouvrir les fichiers de configuration d'un paquet, ou lorsque des fichiers (non marqués comme des fichiers de configuration) ont besoin d'être préservés par `dpkg`, lors de l'installation d'une version plus récente d'un paquet qui contient ces fichiers.

```
# dpkg-divert [--add]  filename # ajoute un détournement
# dpkg-divert --remove filename # supprime un détournement
```

Ne pas utiliser `dpkg-divert` si cela n'est pas absolument nécessaire.

### 3.5.2 Le paquet `equivs`

Si vous compilez un programme depuis les sources, il est mieux d'en faire un paquet Debian (`*.deb`). `equivs` est le dernier recourt.

```
Package: equivs
Priority: extra
Section: admin
Description: Passer outre les dépendances de paquets.
 C'est un paquet qui peut être utilisé pour créer des paquets Debian qui ne
 contiennent que des informations sur les dépendances.
```

### 3.5.3 Commandes de rechange

Pour utiliser `vim` à la place de `vi`, utilisez `update-alternatives` :

```
# update-alternatives --display vi
...
# update-alternatives --config vi
  Selection    Command
-----------------------------------------------
      1        /usr/bin/elvis-tiny
      2        /usr/bin/vim
*+    3        /usr/bin/nvi

Enter to keep the default[*], or type selection number: 2
```

Les commandes de rechange du système sont maintenues dans /etc/alternatives en tant que liens symboliques.

Pour avoir votre environnement de fenêtres X favori, appliquez update-alternatives à /usr/bin/x-session-manager et /usr/bin/x-window-manager.

/bin/sh est un lien direct de /bin/bash ou /bin/ash. Il est plus sûr d'utiliser /bin/bash pour garder la compatibilité avec de vieux scripts bash mais /bin/ash permet d'être plus discipliné et d'être compatible POSIX. La mise à jour vers un noyau Linux 2.4 a tendance à positionner ce lien vers /bin/ash.

### 3.5.4 Utilisation des niveaux de fonctionnement

Lorsqu'ils sont installés, la plupart des paquets Debian configurent leurs services pour être lancés dans les niveaux de fonctionnement 2 à 5. Ainsi, il n'y a pas de différence entre les niveaux 2, 3, 4 et 5 sur un système Debian qui n'a pas été personnalisé ; Debian laisse à l'administrateur local le soin de configurer les niveaux de fonctionnement. Cela diffère de la façon dont les niveaux de fonctionnement sont gérés dans d'autres distributions GNU/Linux populaires. Un changement que vous pouvez vouloire faire est de désactiver xdm ou gdm dans le niveau de fonctionnement 2 pour que X ne soit pas démarré à la fin de la séquence de démarrage ; vous pouvez alors le démarrer en passant au niveau 3.

### 3.5.5 Services désactivés

Les développeurs Debian prennent très au sérieux la sécurité du système. Beaucoup de services sont installés avec le minimum de fonctions activées.

Exécutez ps aux ou consultez le contenu de /etc/init.d/* et /etc/inetd.conf, si vous avez des doutes (à propos d'Exim, DHCP, ...). De plus, consultez /etc/hosts.deny. La commande pidof est aussi utile (voir pidof(8)).

X11 ne permet pas les connexions TCP/IP (distantes) par défaut dans les versions récentes de Debian. La redirection X est aussi désactivée dans SSH.

# Annexe A

# Annexe

## A.1 Auteurs

Guide de référence rapide pour Debian a été démarré par Osamu Aoki <debian@aokiconsulting.com> en tant que mémo d'installation personnel et finalement appelé « Quick Reference … ». Une grande partie du contenu venait des archives de la liste de diffusion « debian-user ». Le « Manuel d'installation Debian » et les « Notes de version Debian » ont aussi été utilisés.

Suite à une suggestion de Josip Rodin, qui est très actif dans le Projet de Documentation Debian (http://www.debian.org/doc/ddp) (DDP) et est le responsable actuel de la « FAQ Debian », ce document a été renommé en « Guide de référence pour Debian » et fusionné avec des chapitres de la « FAQ Debian » ayant un contenu de référence. Par la suite, l'extrait « Guide de référence rapide pour Debian » a été créé.

Ce document a été modifié, traduit et amélioré par les membres de l'équipe QREF suivants :

- Relecture et réécriture en Anglais du « Quick Reference… » original
  - Osamu Aoki <osamu\#at\#debian.org> (leader : tout le contenu)
- Relecture et réécriture en Anglais
  - David Sewell <dsewell\#at\#virginia.edu> (leader : style en)
  - Thomas Hood <jdthood\#at\#yahoo.co.uk> (network related)
  - Brian Nelson <nelson\#at\#bignachos.com> (especially X related)
  - Jan Michael C Alonzo <jmalonzo\#at\#spaceants.net>
  - Daniel Webb <webb\#at\#robust.colorado.edu>
- Traduction en Français
  - Guillaume Erbs <gerbs\#at\#free.fr> (leader : fr)
  - Rénald Casagraude <rcasagraude\#at\#interfaces.fr>
  - Jean-Pierre Delange <adeimantos\#at\#free.fr>
  - Daniel Desages <daniel\#at\#desages.com>
- Traduction en Italien
  - Davide Di Lazzaro <mc0315\#at\#mclink.it> (leader : it)
- Traduction en Portugais (Brésil)
  - Paulo Rogério Ormenese <pormenese\#at\#uol.com.br> (leader : pt-br)

  - Andre Luis Lopes <andrelop\#at\#ig.com.br>
  - Marcio Roberto Teixeira <marciotex\#at\#pop.com.br>
  - Rildo Taveira de Oliveira <to_rei\#at\#yahoo.com>
  - Raphael Bittencourt Simoes Costa <raphael-bsc\#at\#bol.com.br>
  - Gustavo Noronha Silva <kov\#at\#debian.org> (coordinator)
- Traduction en Espagnol (à peu près terminée)
  - Walter Echarri <wecharri\#at\#infovia.com.ar> (leader : es)
  - José Carreiro <ffx\#at\#urbanet.ch>
- Traduction en Allemand
  - Jens Seidel <tux-master\#at\#web.de> (leader : de)
  - Willi Dyck <wdyck\#at\#gmx.net>
  - Stefan Schröder <stefan\#at\#fkp.uni-hannover.de>
  - Agon S. Buchholz <asb\#at\#kefk.net>
- Traduction en Polonais—les membres de PDDP (http://debian.linux.org.pl) suivants :
  - Marcin Andruszkiewicz
  - Mariusz Centka <mariusz.centka\#at\#debian.linux.org.pl>
  - Bartosz Feński <fenio\#at\#debian.linux.org.pl> (leader : pl)
  - Radosław Grzanka <radekg\#at\#debian.linux.org.pl>
  - Bartosz 'Xebord' Janowski
  - Jacek Lachowicz
  - Rafał Michaluk
  - Leonard Milcin, Jr.
  - Tomasz Z. Napierała <zen\#at\#debian.linux.org.pl>
  - Oskar Ostafin <cx\#at\#debian.linux.org.pl>
  - Tomasz Piękoś
  - Jacek Politowski
  - Mateusz Prichacz <mateusz\#at\#debian.linux.org.pl>
  - Marcin Rogowski
  - Paweł Różański
  - Mariusz Strzelecki
  - Krzysztof Ścierski
  - Przemysław Adam Śmiejek <tristan\#at\#debian.linux.org.pl>
  - Krzysztof Szynter
  - Mateusz Tryka <uszek\#at\#debian.linux.org.pl>
  - Cezary Uchto
  - Krzysztof Witkowski <tjup\#at\#debian.linux.org.pl>
  - Bartosz Zapałowski <zapal\#at\#debian.linux.org.pl>
- Traduction en Chinois (simplifié)
  - Hao "Lyoo" LIU <iamlyoo\#at\#163.net> (leader : zh-cn)
  - Ming Hua <minghua\#at\#rice.edu>
  - Xiao Sheng Wen <atzlinux\#at\#163.com> (leader : zh-cn)
  - Haifeng Chen <optical.dlz\#at\#gmail.com>
  - Xie Yanbo <xieyanbo\#at\#gmail.com>
  - easthero <easthero\#at\#gmail.com>
- Traduction en Chinois (traditionnel)

    - Asho Yeh <asho\#at\#debian.org.tw> (leader : zh-tw)
    - Tang Wei Ching <wctang\#at\#csie.nctu.edu.tw> (ex-leader : zh-tw)
- Traduction en Japonais
    - Shinichi Tsunoda <tsuno\#at\#ngy.1st.ne.jp> (leader : ja)
    - Osamu Aoki <osamu\#at\#debian.org>

## A.2 Garanties

Comme je ne suis pas un expert, je ne prétends pas tout connaître sur Debian ou Linux en général. Les considérations de sécurité que j'utilise peuvent être utilisées seulement dans le cas d'une utilisation personnelle.

Ce document ne remplace aucun guide officiel.

Toute garantie est désavouée. Toutes les marques de commerce sont la propriété de leurs propriétaires respectifs.

## A.3 Retour

Les commentaires et additions à ce document sont toujours les bienvenus. Merci d'envoyer un courriel au système Debian BTS (http://bugs.debian.org/) sous le paquet `debian-reference` ou le paquet correspondant à la traduction. L'utilisation de `reportbug` rend facile de remplir un rapport de bogue complet. Vous pouvez toujours envoyer un courriel à Osamu Aoki (http://people.debian.org/~osamu/) <osamu\#at\#debian.org> en Anglais ou à chacun des traducteurs dans leur langue respective.